(4)

NSWC TR 90-167

SELECTIVE LEARNING ALGORITHM FOR CERTAIN TYPES OF LEARNING FAILURE IN MULTILAYER PERCEPTRONS

BY GEORGE ROGERS JEFFREY L. SOLKA
STRATEGIC SYSTEMS DEPARTMENT

JUNE 1990

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

NSWC TR 90-167

SELECTIVE LEARNING ALGORITHM FOR CERTAIN TYPES OF LEARNING FAILURE IN MULTILAYER PERCEPTRONS

BY
GEORGE ROGERS
JEFFREY L. SOLKA
STRATEGIC SYSTEMS DEPARTMENT

JUNE 1990

NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5000 • Silver Spring, Maryland 20903-5000

FOREWORD

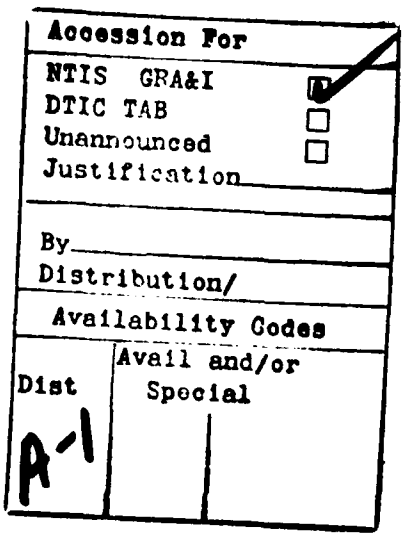
A simple selective learning algorithm for use with Multilayer Perceptrons (MLPs) is presented. This algorithm has proved useful in certain types of problems where learning failure occurs using standard back propagation. Examples of these problems are included. The algorithm is based on the rms output error, computed across all output nodes and all training patterns. The learning rate is decreased for all individual output nodes each time the error is less than a user chosen multiple of the rms error corresponding to the previous pass. This algorithm has produced convergence where the standard fixed gain back propagation failed.

This work has been supported by the Warfare Systems Architecture and Engineering Program and has been conducted in the Space and Ocean Geodesy Branch.

This report has been reviewed by Patrick E. Beveridge, Head of the Space and Ocean Geodesy Branch and J. Ralph Fallin, Head of the Space and Surface Systems Division.

Approved by:

R. L. SCHMIDT, Head
Strategic Systems Department



CONTENTS

INTRODUCTION

The discovery of a back propagation rule for hidden layers in multilayer perceptrons by Werbos,[1] and the subsequent rediscovery by Rumelhart et al.[2] has been fundamental in the resurgence of interest in feed forward networks that has occurred in recent years. Several notable successes by Sejnowski and Rosenberg,[3] Gorman and Sejnowski,[4] and Tesauro and Sejnowski[5] have contributed to the interest and spurred a great deal of optimism in the field. As Minsky and Papert[6] have pointed out however, error back propagation based on gradient descent is not without shortcomings and limitations.

The primary problem we have encountered is that occurring when one of the classes of training patterns is under-represented in the training set. The symptoms of this problem are manifested as one or more patterns for which the output error converges to 1.0 for one or more of the output units. Sometimes this can be circumvented by adjusting the gain and/or momentum term or by varying the network size. In our experience, there are some pathological cases where these measures just do not solve the convergence problem.

A related problem that is perhaps easier to visualize is one where there are a large number (we assume 20 for this example) of output nodes, with each unit corresponding to a distinct class. For each input pattern, the correct output will consist of a single output unit "on" with a value near 1.0, and with the rest "off" with output values near 0. It may happen that there is a local minimum for one set of inputs where all output units are "off." As an example, with 20 output units all "off" with values of zero, the average error is 0.05 which is close to the true minimum error of zero. As the number of output units is increased, the average or rms error decreases so that a local minimum close to the true minimum becomes more likely.

We have developed and used successfully a simple selective learning algorithm which functions to steer the network away from these local minima by distorting the error surface. It is presented in the following section. Some

discussion on its use with two examples are included in the last section along with some closing comments.

SELECTIVE LEARNING ALGORITHM

BACK PROPAGATION RULE

Training the multilayer perceptron is divided into a feed forward step which propagates the input pattern up through the net structure until the output nodes are reached and an error back propagation step which modifies the edge weights of the network to insure improved performance of the network on the next presentation of the pattern.[2] These steps are executed repeatedly for the set of input patterns until the desired answers are obtained at the output nodes. The steps in the feed forward process are as follows. For the j-th node a weighted sum of its inputs is computed

$$T_{pj} = \sum_i w_{ji} o_{pi} + \theta_j$$

In the next step the sigmoid transfer function is applied to this weighted sum

$$o_{pj} = S(T_{pj}) = \frac{1}{1 + \exp(-\alpha T_{pj})}$$

In the back propagation phase, edge weight corrections are computed. First the error for each output node j is computed

$$\delta_{pj} = (t_{pj} - o_{pj}) o_{pj} (1 - o_{pj})$$

Next the error term for each hidden unit j is computed.

$$\delta_{pj} = o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} w_{kj}$$

The δ_{pk}s are the error contributions from the nodes above node j in the network. Once all the error terms are computed the edge weights are adjusted according to

$$w_{ji}(t + 1) = w_{ji}(t) + \varepsilon \delta_{pj} o_{pi} + \eta (w_{ji}(t) - w_{ji}(t - 1))$$

Weights are adjusted after each presentation of each pattern. The offsets for the hidden and output units are treated as edge weights from units which have a constant value of 1.

SELECTIVE LEARNING FOR BACK PROPAGATION

The fundamental idea of our algorithm is to selectively deweight the output errors for certain output units in a dynamic manner. This is accomplished by multiplying the output error by a constant (<1) for those output units for which the output error is less than a user input multiple of the rms error σ as computed based on all of the output units for all training patterns.

The two user inputs are:

$\xi \equiv$ scaling factor for the output error term,

$\rho \equiv$ multiple of the rms error.

The steps in the algorithm are:

1. For the first pass through all of the training patterns, the output error for all output units is decreased by ξ:

   $$(t_{pj} - o_{pj}) \rightarrow \xi(t_{pj} - o_{pj}), \quad \forall p,j.$$

   This effectively decreases the gain for all corrections by a factor of ξ.
2. All weight and offset adjustments are carried out as ususal (see The Back Propagation Rule).
3. The rms error σ is computed for the current pass.
4. The next pass is carried out with selective learning. For each output unit for each pattern;

   $$\forall p,j: \text{ if } |t_{pj} - o_{pj}| < \rho\sigma, \text{ then}$$
   $$(t_{pj} - o_{pj}) \rightarrow \xi(t_{pj} - o_{pj}).$$

5. All weight and offset adjustments are carried out as usual based on the reduced output errors.
6. Repeat starting at step 3.

APPLICATION STRATEGIES

There is generally no point in employing selective learning until a training problem has been encountered. Once it is clear that the problem is not simply too small a network, then selective learning should be considered as a means of solution. An approach that has been found to work very well is to start with a "brain" or set of connection weights and offsets developed using standard back propagation that has a maximum error of ≈ 0.9. The parameter ρ can then be chosen so as to reduce the error terms for those output units whose errors are less than ≈ 0.8. The goal is to set ρ so that only the few worst outputs errors are unreduced. By increasing ρ, it is possible to reduce learning for all but one of the output units on one of the training patterns as an extreme example. A good rule of thumb for an initial value of ξ is either the ratio of problem training patterns to good training patterns, or the inverse of the number of output units in the case where only a single output unit per training case is to be on. In either case, it is often necessary to tweak the value somewhat. As the maximum error decreases, σ may increase or decrease, and it is usually necessary to occasionally modify ρ.

RESULTS AND CONCLUSIONS

This selective learning algorithm has been successfully applied to training a connectionist expert system and to radar data processing with impressive results.

In the connectionist expert system application,[7] the training set consisted of 722 input patterns or rules, with a network configuration of 23 input units, 104 and 52 units in the first and second hidden layers, respectively, and 18 output units. Each output unit corresponded to a particular decision. Thus only a single output unit should be on, all others being off. Since each of the input patterns represented a rule from the 722-rule rule base, it was unacceptable to have any of the training patterns learned incorrectly. Also, it was not possible to assure that all classes of rules be adequately represented in the training set. Several of the classes were underrepresented, leading to learning problems with the standard back propagation.

No amount of changing the network size or the values of the gain or momentum parameters resulted in a network that could successfully learn all of the rules. There were always at least two failures. In an expert system of any type, this is clearly unacceptable. With the selective learning employed, the network was able to successfully learn all of the training patterns.

The other major application for which we have used the algorithm is in the processing of 3-dB signal/noise radar data.[8] It has been used to boost performance in learning the training set of 1600 input data sets consisting of 160 input values each. It has proven to be the only method of successfully learning all of the training patterns. (Expectedly, this decreases the generalization performance.) It has also been used to reduce training times by a factor of five when the goal was to achieve the same training results as with standard back propagation.

In summary, the described selective learning algorithm has proven useful in cases where standard error back propagation fails. It is probably most useful where accurate memorization of training patterns is required. In cases such as the radar signal processing, where it is not desirable to memorize all of the noisy training data and over training can occur, its utility is much less clear cut. Nonetheless, it resulted in faster training with comparable generalization results when employed as a training accelerator in the radar case.

GLOSSARY

o_{pj} = Output of the j-th node on the p-th pattern

q_j = Offset of the j-th node

S = Sigmoid function

t_{pj} = Desired output of the j-th mode on the p-th pattern

T_{pj} = Intermediate result for the j-th node on the p-th pattern

w_{ji} = Weight from i-th node in layer n to j-th node in layer n + 1

α = Sigmoid slope parameter

ε = Learning rate

η = Rate of momentum transference

ξ = Scaling factor for the output error term in selective learning

ρ = Multiple of the rms error used in selective learning

REFERENCES

1. Werbos, P. J., *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph.D. thesis, Harvard University Committee on Applied Mathematics, November 1974.

2. Rumelhart, D. E., McClelland, J. L., and the PDP research group, *Parallel Distributed Processing Vol. 1: Explorations in the Microstructure of Cognition*, (1986) MIT Press.

3. Sejnowski, T. J., and Rosenberg, C. R., "Parallel networks that learn to pronounce English text.," *Complex Systems*, 1, pp. 145-168 (1987).

4. Gorman, R. P., and Senjowski, T. J., *Neural Networks*, 1 (1988) pp. 5-89.

5. Tesauro, G. and Sejnowski, T. J., "A Parallel Network that Learns to Play Bacgammon," Submitted to Artificial Intelligence.

6. Minsky, M. and Papert S., *Perceptrons*, Cambridge, MA: MIT Press.

7. Rogers, G. W., Solka, J. L., and Steffen, D., "A Neural Network Implementation of an F-14 Battle Management Fusion Algorithm Rule Base," Submitted to *Simulation*.

8. Solka, J. L., Rogers, G. W., and Harrell, T. J., "Signal Processing with Neural Networks: I. Signal in Noise," in preparation.

DISTRIBUTION

	Copies
Commander	
Space and Naval Warfare	
Systems Command	
Attn: Code 301	1
Code 3011	1
Code 311	1
Washington, DC 20363-5100	
Defense Advanced Research	
Projects Agency	
Attn: DSO (Yoon)	1
1400 Wilson Blvd.	
Rosyln, VA 22201	
Commander	
Naval Weapons Center	
Attn: Code 3903 (Andes)	1
China Lake, CA 93555	
Defense Technical Information	
Center	
Cameron Station	12
Alexandria, VA 22314	

	Copies
G71 (Gray)	1
K	1
K105	5
K10	5
K12	5
K12 (Roger)	10
K12 (Solka)	10
K13 (Shuler)	5
K14	5
K14 (Steffen)	10
K44 (Glass)	2
K52 (Farr)	2
N05	1
N06	1
N13	1
N24	1
N24 (Bailey)	1
N35 (Kuchinski)	2
N415	2
R04	1
U31	1

Internal Distribution:

C	1
D	1
D4	1
D24 (Bailey)	1
D25	1
E231	3
E232	2
E32 (GIDEP)	1
F41 (Kruger)	1
FOX	1
G07	1
G12	1
G42 (Farsaie)	1

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 1990	Final

4. TITLE AND SUBTITLE
SELECTIVE LEARNING ALGORITHM FOR CERTAIN TYPES OF LEARNING FAILURE IN MULTILAYER PERCEPTRONS

5. FUNDING NUMBERS

6. AUTHOR(S)
George Rogers and Jeffrey L. Solka

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Surface Warfare Center (K12)
Dahlgren, VA 22448-5000

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWC TR 90-167

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A simple selective learning algorithm for use with Multilayer Perceptrons (MLPs) is presented. This algorithm has proved useful in certain types of problems where learning failure occurs using standard back propagation. Examples of these problems are included. The algorithm is based on the rms output error, computed across all output nodes, and all training patterns. The learning rate is decreased for all individual output nodes each time the error is less than a user chosen multiple of the rms error corresponding to the previous pass. This algorithm has produced convergence where the standard fixed gain back progagation failed.

14. SUBJECT TERMS
selective learning algorithm, Multilayer Perceptrons (MLPs), standard back propagation, rms output error,output nodes,standard fixed gain back propagation

15. NUMBER OF PAGES
13

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR